Creating Legible Robotic Motion via Local Planning

A Thesis

Presented to

The Academic Faculty

By

Dijon Stowers

In Partial Fulfillment

Of the Requirements for the Degree in Computer Science in the

College of Computing, Georgia Institute of Technology

Georgia Institute of Technology

Spring 2018

Abstract:

A requirement for human robot collaboration is that the robot's movements display intent early in the interaction so that a human may respond to the action appropriately. Regarding autonomous navigation, local planning is responsible for creating this motion relative to a global plan in an environment with dynamic obstacles. This research is the augmentation, implementation, and testing of ROS embedded local planners DWA and TEB for the purpose of creating legible motion

Introduction

As robots become more accessible in the home environment, it becomes important that a robot's navigation pathing be legible for humans the robot is commonly interacting with [1]. Human reactionary movement is largely based on the perceived intent of the movements of the obstacle or entity the human is attempting to overcome. Making functional movement less a necessity when interacting with people as intent representation becomes the dominant requirement [2]. Human agents are not static, and so to produce legible motion in a practical human filled environment, what is responsible for overcoming obstacles is here augmented. More specifically, local planners interpret dynamic movements of the environment and create a pathing to handle the detected objects, this research considers the augmentation of local planners to produce legible object avoidance, which per a global plan produces legible paths. Then in testing the augmented paths, first in simulation and then physically, a relative measure of legibility be determined.

Functional vs Legible Motion

Functional motion is defined as motion that prioritizes efficiency where in legible motion attempts illustrate intent by orienting a robot's physical model towards goal earlier in execution of an action [2], which may yield suboptimal execution times, but the intent is more easily communicated

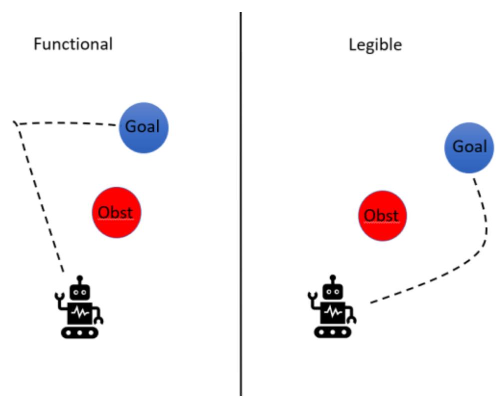

Figure 1. This displays a representation of the differences between functional and legible motion

Methods and Materials

The two algorithms considered are the Dynamic Window Approach (DWA) and the Timed Elastic Band (TEB) planners paired with the base navigation stack global planner all built within the ROS middleware

Dynamic Window Approach

The Dynamic Window Approach (DWA) provides a sample-based optimization by predicting several different possible velocity movements within the robot's control space relative to a global path and produces a local grid map for which the samples are generated After throwing out illegal movements the DWA selects the best trajectory at a given velocity defined by a cost function that assesses distance from global path, distance to goal and obstacle avoidance for some predefined step size [3]. The onboard controller then translates to determine directional velocities to produce the found "best" trajectory. By manipulating the values of the cost function and deciding pathing tolerance for the DWA for what is and is not valid, thus creating different means of reaching a goal [4], possibly more legible. Because the DWA is easy to augment and is computationally quick, through simple augmentation of its core cost function, more legible paths based around object avoidance and goal progression, more legible consideration for path generation should be prioritized in implementation.

Timed Elastic Band

The Timed Elastic Band planner provides a continuous optimization solution to local planning. By sampling a subset of the subscribed global plan within the local cost map, a local goal is defined. By default, TEB optimizes out one path that takes the least amount of time and executes that found path at each time interval of the found trajectory. To prevent the algorithm from getting fixed on one solution for any optimal pathing, other admissible paths are also simultaneously generated using a version of homotopy along several trajectories among different topologies [5] which, by varying the distance at each time interval and then running the same optimization function over the generated new paths, thus providing several solutions and preventing becoming fixed. Which in terms of legibility should better account for dynamic objects. The cost function deciding the optimality depends on the Levenberg-Marquardt algorithm to create a best fit curve to the constraint approximations specifically defined by constraints and predefined weights. Given that this optimization forces a relatively smooth curve and legible actions also resemble smooth curves [6], use of TEB to create a curve around obstacles relative to a global path is the intuition in using it for increasing legibility.

Procedure

Beginning with the implementation of DWA, the central cost function determines the behavior of the algorithm. The cost function is the following equation

$Cost = (w_1 * Pd) + (w_2 * Gd) + (w_3 * Od/100)$

Where Pd represents the path deferment from the last received trajectory relative to the global plan and scores higher paths that do not defer much. Goal distance denoted Gd in the function, is the distance from the goal by any trajectory produced scoring higher trajectories that end closer to the goal. Lastly is the variable Od which represents the object tolerance for any trajectory. Each of these variables are multiplied by a weight whose relationship to the other weights determines the behavior of the algorithm [7]. Therefore, in comparing weight relationships before attempting to find optimal weights, a clear domain of weight distribution becomes evident on which to begin searching for the most legible behavior. Through augmenting the weights such that behavior was noticeably different when running a simulated Fetch in an environment with static obstacles, the following domain classes were used as denoted in. It is also important to note that because the assumed core of creating legible motion in local planning is obstacle avoidance [8], the Od variable is never lowered beyond its default value as it is the variable that scores the behavior that needs to be exhibited.

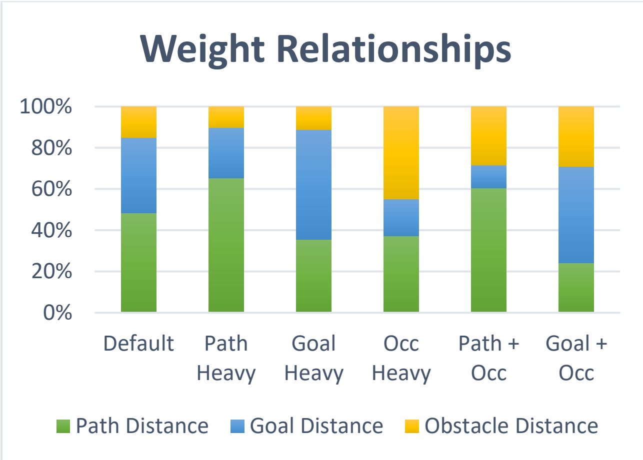

Figure 2. A relational graph denoting the amount of weight given to the input variables of the algorithm

Relational Behavior

The default configuration prioritizes taking paths that are closer to the path already traveled and only considers object avoidance and path deferment when it becomes necessary to continue progressing towards the goal [9]. Increasing the weight on the path distance variable gave an exaggerated result of the typical trajectories generated by the default configuration, but instead occasionally became stuck and obstacles that were able to be overcome with the default weights now were insurmountable. Similarly, with increasing the allowance for goal tolerance, Fetch would often get stuck as within any iteration close to an obstacle where the default would have disqualified, became a valid a trajectory, this configuration marks it as valid because the goal of the algorithm then becomes moving towards the goal above all else. Finally forcing the algorithm to better avoid obstacles yields smoother curves, but with too many static obstacles in a simulated environment, a metaphysical wall is built making all trajectories invalid as the robot cannot produce a trajectory outside of the object tolerance.

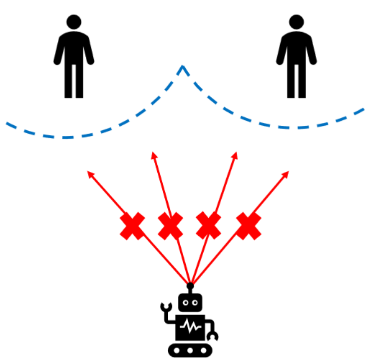

Figure 3. A depiction of the metaphysical wall problem, notice that there exists a “barrier” around the humans as they are recognized as objects.

Taking the extreme cases and combining them with lowered object tolerance is represented by the last two combined domains in the figure. Dampening the weight on goal tolerance and subsequently increasing the weights on obstacle avoidance and path deferment, in simulation, the algorithm continues to the goal, but begins running into the problem of a metaphysical wall. Using a similar configuration, bu instead prioritizing simply moving towards the goal along with increased object avoidance, yielding curved paths around obstacles that move towards the goal, but not is easily perceived as legible because path direction changes heavily from each step in the iteration to the next, given the variable controlling path distance from the previous iteration is decreased.

Timed Elastic Band

By default, the TEB generates a smooth, relatively legible curve.

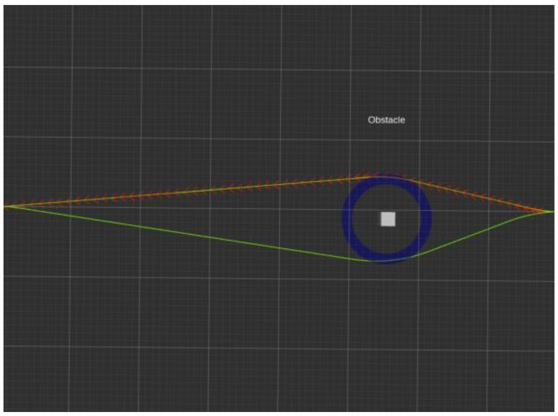

Figure 4. A subgoal of the global path within RViz created using default TEB. The top path is shown as the preferred time optimal path.

The paths generated similarly to the DWA by means of a cost function, however TEB performs a best fit via the following formula for perceived obstacles in any time step.

$$b^* = \sum_i \sigma_i f_i^2(b)\,, i \in \{J, P\}$$

Note that the notation b* represents the optimal pathing and is equal to the aggregate of non-linear least squares constrained by objectives denoted by J and penalties P and then multiplied by some weight σ_i. The optimal discretized trajectory b^* is obtained by my minimizing the cost function while still attempting to accurately represent the objectives and penalties specified by parameters defined within the algorithm [10]

With TEB the individual weights are left default while the bounds of the constraints are instead changed which allows for different pathing behavior. The augmented constraint values are increased inflation distance, the buffer zone around obstacles, and keeping this above the minimum obstacle distance, which attempts to mandate a minimum curve around obstacles. Distinctive paths as shown in the following figure are generated.

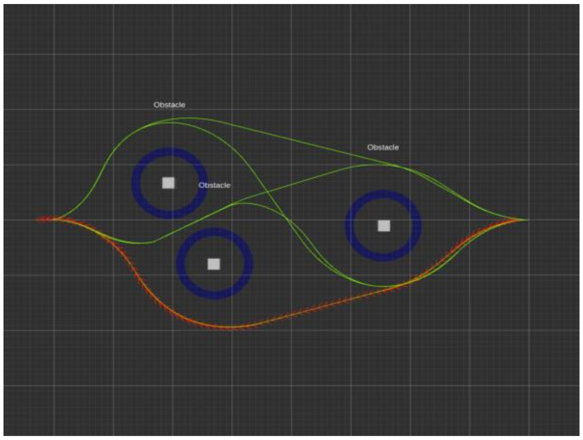

Figure 5. This displays how the augmented TEB allowing for more curved legible paths compared Figure 4 within simulation

Physical Implementation

When faced with the physical dynamic obstacles in a real environment, the DWA algorithm is unable to overcome the issue of the metaphysical wall experienced when the obstacle bias dominated the weight distribution regardless of any tangible increase in consideration for object avoidance. Fetch is able to consistently reach a goal pose when running TEB as the local planner, being that TEB continues to search for a solution despite it's given constraints.

Findings

In simulation, using the Dynamic Window Approach is relatively effective in increasing legible motion as curved paths can be the forced preference, however because the curved paths are made inconsistent by the path deferment weight being decreased, the curved paths may be generated in changing directions when faced with several static objects. Dynamic objects in the physical implementation were insurmountable making the implementation DWA infeasible to implement for the purpose of increased legibility.

The Timed Elastic Band local planner generates more legible paths as it, by construct, generates smooth curves which is natively more representative of legible motion. In physical motion

DWA nor TEB are built for legibility and so forcing parameters for a desired use that the planner is not built for yields suboptimal results. Local planning focuses on object avoidance which enables the metaphysical wall problem. Because TEB creates a subset of the subscribed global plan, if the algorithm is instead configured not for only optimality, but solely generating legible curves from the beginning pose to the subset goal, a more consistent pathing could be generated. Thus, creating a legible planner rather than repurposing others.

References

[1] Dragan, A. D., Bauman, S., Forlizzi, J., & Srinivasa, S. S. (2015). *Effects of Robot Motion on Human-Robot Collaboration*. Retrieved 12 6, 2018, from https://dl.acm.org/citation.cfm?id=2696473

[2] Stulp, F., Grizou, J., Busch, B., & Lopes, M. (2015). *Facilitating intention prediction for humans by optimizing robot motions*.

[3] *ROS/dwa_local_planner - ROS Wiki*. (n.d.).

[4] Fox, D., Burgard, W., & Thrun, S. (1996). *Controlling synchro-drive robots with the dynamic window approach to collision avoidance*.

[5] *ROS/teb_local_planner - ROS Wiki*. (n.d.).

[6] Koubaa Anis, *Robot Operating System (ROS): The Complete Reference, Volume 3*: Springer 2018

[7] Rosmann, C., Hoffmann, F., & Bertram, T. (2017). *Online Trajectory Planning in ROS Under Kinodynamic Constraints with Timed-Elastic-Bands*.

[8] Dragan, A. D., & Srinivasa, S. S. (2013). *Generating Legible Motion*.

[9] Sisbot, E. A., Marin-Urias, L. F., Alami, R., & Siméon, T. (2007). A Human Aware Mobile Robot Motion Planner. *IEEE Transactions on Robotics, 23*(5), 874-883.

[10] Marin-Plaza, P., Hussein, A., Martín, D., & Escalera, A. d. (2018). Global and Local Path Planning Study in a ROS-Based Research Platform for Autonomous Vehicles. *Journal of Advanced Transportation, 2018*, 1-10.